Taxi-v3 with Dynamic Programming (DP) — Short Tutorial

Policy Evaluation · Policy Iteration · Value Iteration (Gymnasium)

This short written tutorial explains how to run the project, tune hyperparameters, understand the code pipeline, and interpret the main results and plots. It doubles as a script for a ≤10-minute video walkthrough.

1) What you'll see

- Dynamic Programming (DP) on Gymnasium's Taxi-v3 using exact model-based backups: Policy Evaluation, Policy Iteration, and Value Iteration.
- How to execute the script, adjust hyperparameters, and read the outputs: convergence logs, evaluation metrics, value-function heatmap, and greedy-policy arrows.

2) Requirements & Installation

Python 3.9+ and the following packages:

```
pip install "gymnasium[toy_text]" numpy matplotlib pandas pygame
```

Conda alternative:

```
conda install -c conda-forge gymnasium pygame numpy matplotlib pandas
```

3) How to Run

Place your script (e.g., TaxiV3DP.py) in the project folder and run:

```
python TaxiV3DP.py
```

You will see: (i) convergence logs for PI and VI; (ii) evaluation metrics (avg reward, std, success, avg steps); (iii) plots (bar charts, convergence curves, heatmap, policy arrows); and (iv) a results pickle file taxi_dp_results.pkl.

4) What the Script Does

- Creates the Taxi-v3 environment and reads the tabular transition model P via env.unwrapped.P (probability, next state, reward, terminal).
- Runs Policy Iteration (PI): iterative Policy Evaluation (Bellman expectation backups) + Greedy Policy Improvement (argmax over one-step lookahead Q).
- Runs Value Iteration (VI): Bellman optimality backups on V until the residual Δ = ||V_{k+1} − V_k||_∞ < θ, then greedy policy extraction.

- Evaluates each learned policy over multiple episodes; prints Avg Reward, Std Reward, Success Rate, and Avg Steps.
- Generates visualizations: bar charts (reward, success, convergence iterations, steps), VI/PI convergence curves, value-function heatmap (fixed passenger/destination slice), and greedy-policy arrows.

5) Hyperparameters

Key knobs you may tweak and their effects:

```
• gamma (default: 0.99): Discount factor. Higher means the value looks
further into the future; convergence may be slightly slower.
• theta (default: 1e-6): Convergence tolerance on the max-norm residual
Δ. Smaller means higher precision, more iterations.
• max_iterations (e.g., 1000): Safety cap for VI/PE sweeps.
• num_episodes (e.g., 10–100): Number of test episodes; more → lower
variance in reported metrics.
• render (True/False): If True, shows a quick animation for the first
few episodes.
• max_steps (default: 200): Per-episode time limit.
```

Optional stochastic environment flags:

```
• is_raining=True: Movement noise (0.8 intended, 0.1 left, 0.1 right).
If a branch hits a wall, it becomes a noop but counts in expectation.
• fickle_passenger=True: After the first pickup and moving one square
away from the source, the passenger changes destination with 30%
probability (one-time).
```

6) Core Equations (Bellman)

```
Bellman expectation (Policy Evaluation):
V^π(s) = Σ_a π(a|s) Σ_{s',r} p(s',r|s,a) [ r + γ V^π(s') ]

Greedy improvement (Policy Iteration):
Q^π(s,a) = Σ_{s',r} p(s',r|s,a) [ r + γ V^π(s') ],   π_new(s) ∈
argmax_a Q^π(s,a)

Bellman optimality (Value Iteration):
V^*(s) = max_a Σ_{s',r} p(s',r|s,a) [ r + γ V^*(s') ],   π^*(s) ∈
argmax_a Σ_{s',r} p(·)[ r + γ V^*(s') ]

Stopping criterion:
Δ_k = ||V_{k+1} − V_k||_∞ < θ,   with error bound  ||V_k − V^*||_∞ ≤
(γ/(1−γ)) Δ_k
```

7) Configuration & Flags

```
Change discount/tolerance:
solver = TaxiDynamicProgramming(gamma=0.99, theta=1e-6)

Use stochastic model for exact DP (refresh P):
solver.env = gym.make('Taxi-v3', is_raining=True,
fickle_passenger=True)
solver._build_transition_model()  # refresh self.P from this env

Reproducibility (testing seeds):
state, _ = test_env.reset(seed=42)
import numpy as np; np.random.seed(42)
```

8) Interpreting Outputs

- Convergence logs: VI typically crosses θ = 1e-6 in ~19 sweeps (geometric-like decay due to contraction). PI requires few outer policy improvements; first PE from random may be long, subsequent ones ~19 sweeps.
- Results table: Convergence (iterations), Avg Reward, Std, Success Rate (%), Avg Steps. On Taxi, (no illegal actions) Return ≈ 21 − Steps.
- Plots: Bars (reward/steps/success/iterations), VI residual curve (semilog), PI evaluation residual (concatenated), value heatmap (fixed passenger/destination), and greedy-policy arrows.

9) Optional Experiments

- Stochastic (train & test): Expect higher Avg Steps, lower Avg Reward, higher variance; Success may drop if time limit is tight.
- Mismatch (train deterministic, test stochastic): Shows model sensitivity — performance degrades relative to deterministic baseline.

10) Closing Notes

In tabular MDPs with an exact model, both PI and VI converge and yield near-perfect policies on Taxi-v3. Differences in reported iterations reflect different units (VI sweeps vs. PI outer improvements). Visualizations (convergence, heatmap, arrows) help explain both convergence behavior and spatial policy structure.